CLAIMS

We claim:

1. A driver system interface, comprising:

an operating system (OS) interface to process a plurality of messages for a plurality of internal driver entities ; and

a message controller coupled to the OS interface to transfer the plurality messages.

2. The driver system interface of claim 1, further comprising:

a platform interface coupled to the message controller to provide platform specific information to the message controller.

3. The driver system interface of claim 1, wherein the message controller communicates with the OS interface through functions.

4. The driver system interface of claim 1, further comprising:

a plurality of message channels to communicate the plurality of messages to and from the plurality of internal driver entities.

5. The driver system interface of claim 4, the message controller comprising:

a plurality of installable components corresponding to the plurality of message channels

6. The driver system interface of claim 5, wherein the plurality of installable components comprise function pointers corresponding to functions in the OS interface.

7. The driver system interface of claim 1, wherein the message controller routes the plurality of messages to a plurality of internal entities.

8. The driver system interface of claim 1, the OS interface comprising:

an external interface to communicate with the plurality of external driver entities.

9. The driver system interface of claim 1, wherein each message of the plurality of messages comprises a message header portion containing routing information for the message controller and a message information portion containing data related to an action for a target entity to perform.

10. The driver system interface of claim 9, wherein the message header portion comprises an event variable to indicate a unique event for a corresponding message channel and a message channel identifier variable to indicate the corresponding message channel.

11. A communications driver comprising:

a network driver interface; and

a miniport driver coupled to the network driver interface, the miniport driver comprising:

a system interface abstraction layer (SIAL) comprising:

an operating system (OS) interface to process a plurality of messages for a plurality of internal driver entities; and

a message controller coupled to the OS interface to transfer the plurality of messages.

12. The communications driver of claim 11, the SIAL further comprising:

a platform interface coupled to the message controller for providing platform specific information and commands to the message controller.

13. The communications driver of claim 11, wherein the message controller communicates with the OS interface through functions.

14. The communications driver of claim 11, the message controller further comprising:

a plurality of message channels, each message channel for communicating a subset of the plurality of messages to and from a corresponding subset of the plurality of internal devices to a specific external device.

15. The communications system driver of claim 14, wherein the message controller comprises a plurality of installable components corresponding to the plurality of message channels.

16. The communications system driver of claim 15, wherein the plurality of installable components comprise function pointers corresponding to functions in the OS interface.

17. The communications driver of claim 11, the OS interface comprising:

an external interface for communicating with the plurality of external entities.

18. The communications system driver of claim 11, the network driver interface further comprising:

a dynamic messaging library coupled to the SIAL.

19. The communications system driver of claim 11, wherein each message of the plurality of messages comprises a message header portion containing routing information for the message controller and a message information portion containing data related to an action for a target entity to perform.

20. The communications system driver of claim 19, wherein a message header comprises an event variable to indicate a unique event for a corresponding message channel and a message channel identifier variable to indicate the corresponding message channel.

21. A communications card, the communications card comprising:

a communications system driver comprising:

a network driver interface;

a miniport driver coupled tot he network driver interface; and

a system interface abstraction layer (SIAL) coupled to the network driver interface and the miniport driver, the SIAL comprising:

an operating system (OS) interface for processing a plurality messages to and from a plurality of entities internal to the OS; and

a message controller coupled to the OS interface for translating the messages and routing the message to and form an entity external to the OS.

22. The communications card of claim 21, the SIAL further comprising:

a platform interface coupled to the message controller for providing platform specific information and commands to the message controller.

23. The communications card of claim 21, wherein the message controller communicates with the OS interface through functions.

24. The communications card of claim 21, the message controller further comprising:

a plurality of message channels, each message channel for communicating a subset of the plurality of messages to and from a corresponding subset of the plurality of internal devices to a specific external device.

25. The communications card of claim 24, wherein a message header comprises an event variable to indicate a unique event for a corresponding message channel and a message channel identifier variable to indicate the corresponding message channel.

26. The communications card of claim 24, wherein the message controller comprises a plurality of installable components corresponding to the plurality of message channels.

27. The communications card of claim 26, wherein the plurality of installable components comprise function pointers corresponding to functions in the OS interface.

28. The communications card of claim 21, the OS interface comprising:

a external interface for communicating with the plurality of external entities.

29. The communications card of claim 21, the communications card further comprising:

a dynamic messaging library coupled to the SIAL.

30. A communications driver, comprising:

a network driver interface; and

a driver system interface comprising:

an external interface to communicate with a plurality of external driver entities; and

an internal interface to communicate with a plurality of internal driver entities.

31. The communications driver of claim 30, wherein the external interface handles the semantics of the plurality of external driver entities.

32. The communication driver of claim 30, wherein the external interface is a portion of an operating system (OS) interface.

33. The communication driver of claim 30, wherein the internal interface comprises a message controller to control a plurality of message channels to pass a plurality of messages between the plurality of external driver entities and the plurality of internal driver entities.

34. A method of abstracting a driver system interface, the method comprising the steps of:

creating a platform specific and operating system specific message channel between an internal driver entity and an external driver entity; and

routing a software message between the internal driver entity and the external driver entity through the platform specific and operating specific message channel.

35. The method of claim 34, further comprising the steps of:

creating a plurality of platform specific and operating specific message channels between a plurality of internal driver entities and a plurality of external driver entities; and

routing a plurality of software messages between the plurality of internal driver entities and the plurality of external driver entities.

36. The method of claim 34, further comprising the steps of:

creating a platform specific and operating specific message channel between a first internal driver entity and a second internal driver entity; and

routing a software message between the first driver entity and the second driver entity through the platform specific and operating system specific message channel.

37. The method of claim 34, wherein the routing step is performed by an installable component corresponding to the message channel.

38. The method of claim 34, wherein the software message comprises a header portion containing routing information and an information portion containing data specific to an action to be performed by a target driver entity.

39. The method of claim 34, wherein the internal driver entity is a miniport driver module and the external driver entity is a non-miniport driver module.